Hardening Ubuntu

Date: 12 Mar 2011

Author: Jonathan Marsden jmarsden@fastmail.fm

URL: http://crosswire.org/~jmarsden/talks/hardening-ubuntu/hardening-ubuntu.html

Contents

• Introduction
• The BASICS (the bare minimum)
• Specific security-oriented changes
• Checking your server
• Manage (and read) your logs
• Further Reading

Introduction

• There are bad people (and bad bots) out there
• What can you do to improve the odds of your server surviving?
• Security is a process, not a one-time fix
• Increased security often means decreased convenience!

The BASICS (the bare minimum)

Know where your software comes from

• Only use packaged software in official repositories
• Installing "newer better faster" stuff from tar.gz or from a PPA or from a downloaded .deb means trouble.

Keep your system up to date

sudo apt-get update && sudo apt-get upgrade

• Consider apticron to email you a list of updates
• Consider unattended-updates to do them for you

When kernels change or new packages are needed

sudo apt-get update && sudo apt-get dist-upgrade
Remove unnecessary software

- The less software is on your machine the less bugs are on your machine
- Bad guys can't exploit holes in software you do not install

Don't run services you do not need

- If you need services for internal use only, bind them to localhost

Check what is listening:

```
sudo netstat -ntlp4   # for IPv4 services
```

Make backups!

- Keep them offline so they cannot be tampered with
- Automate them so they really do happen
- Test them once in a while (quarterly?)
- Many options for how to backup: rsync over ssh is one

Put the machine behind a firewall router

- You can put Ubuntu server directly on the Internet
- If you have the option, use a router -- one more layer of protection

If you think you "were hacked"

- Disconnect the machine from the Internet
- Get expert help ASAP

Specific security-oriented changes

Use UFW for basic packet filtering

- UFW is an iptables-based firewall, easy to configure and use
- See https://help.ubuntu.com/community/UFW

Example ufw commands:

```
sudo apt-get install ufw
sudo ufw enable
sudo ufw logging on
sudo ufw show verbose
sudo ufw status
```
sudo ufw allow ssh  # May be done automatically by packages these days
sudo ufw allow http
sudo ufw default deny

- Only open ports you actually use!
- If you know enough to want to do clever firewall tricks, ufw is not for you

**Denyhosts: block most SSH attacks**

- Blocks people who repeatedly try and fail to login using SSH
- Adds their IPs to /etc/hosts.deny

  sudo apt-get install denyhosts

- No further configuration is needed (nice!)

**Secure Shared Memory**

- /dev/shm can be used in an attack against a running service, such as httpd.

Modify /etc/fstab to make it more secure:

  sudoedit /etc/fstab
tmpfs /dev/shm tmpfs defaults,noexec,nosuid 0 0

- Consider making it read only (ro) and no devices (nodev) -- but TEST if you do that
- Google Chromium (browser) breaks if /dev/shm is ro

**Disable root SSH login**

- The root account is disabled by default in Ubuntu.
- If you installed Ubuntu on a VPS like Slicehost or Linode, root is enabled.
  - In any case, it is a good idea to disable root SSH access.

Edit /etc/ssh/sshd_config and set PermitRootLogin to no:

  sudoedit /etc/ssh/sshd_config

Change PermitRootLogin to no:

  PermitRootLogin no

- Restart sshd

Consider limiting which users can use ssh on a multiuser server
Only allow admin users to see and use su and sudo

Helps prevent privilege escalation.

```bash
sudo dpkg-statoverride --update --add root admin 4750 /bin/su
sudo dpkg-statoverride --update --add root admin 4750 /usr/bin/sudo
```

Kernel configuration: Deny source routing of incoming packets

- Ubuntu now has sane defaults here

Possible commands:

```bash
sudo sysctl -w net.ipv4.conf.all.accept_source_route=0
sudo sysctl -w net.ipv4.conf.default.accept_source_route=0
```

- More generally, read `/etc/sysctl.conf` and `/etc/sysctl.d/*`
- If you fully understand things, consider changes for improving security!

Deny system users access to FTP

- If ftplib is installed and running.
- Removes a FAIL from the tiger report (see later).
- SFTP (the SSH-based one) is more secure than FTP, use it instead if possible.

Edit `/etc/ftpusers`: add system users to deny use of ftplib:

```bash
for i in backup bin daemon games gnat irc libuulid list lp \   mail man mysql news ntp postfix proxy sshd sync sys syslog \   uucp www-data
do
echo $i |sudo tee -a /etc/ftpusers
done
```

Checking your server

**nmap - scan yourself**

```bash
sudo apt-get install nmap
sudo nmap -v -ss localhost
```
• Looks for open ports, where are things listening
• Do not run nmap remotely against other people's servers!
• On a workstation, use zenmap for a GUI front end

tiger - host config and permissions checks

• An old host-based security checker
• Still worth running
• Runs daily from cron
• Best if tailored to your system and needs in /etc/tiger/tigerrc

chkrootkit - check for rootkits

```bash
sudo apt-get install chkrootkit
sudo chkrootkit
```

• Imperfect, but better than nothing
• There are also rkhunter and unhide

aide (or tripwire) - what files changed?

• Takes work to set up and tweak
• Will tell you which files were changed (or added or deleted)
• Used for /etc and /usr can be very helpful after a mild hack
• Keep the checksum database offline (so it can't be tampered with)

Manage (and read) your logs

```bash
sudo apt-get install logwatch
sudo logwatch |less
```

• logwatch - useful log file analyzer
• swatch - alert on specific log entries

• However you do it, read your logs!

Further Reading

• https://help.ubuntu.com/community/StricterDefaults - banner, ssh restriction, etc.
• http://www.linuxsecurity.com/
• http://www.amazon.com/Linux-Firewalls-Detection-Response-iptables/dp/1593271417/